Programação Orientada a Objecto

**Paradigma de Orientação a Objecto**

《 LukényaCristóvão/1000021135》

Faculdade de Engenharia

Universidade Católica de Angola

《[brcristovao69@gmail.com](mailto:brcristovao69@gmail.com) 》

**Abstract-- Um resumo muito conciso sobre o método de Paradigma de orientação á objecto**

Resumo:

**O Paradigma de Orientação a Objecto (POO) trás uma ideia muito interessante a reapresentação de cada elemento em termos de um objeto ou classe, permite aproximar um sistema que está a ser criado ao que é observado no mundo real.**

***Palavras chaves - 《Métodos, Classes, Propriedades e Objetos》***

INTRODUÇÃO

Neste trabalho vimos que o POO é baseado no conceito de objetos que podem conter dados de forma de campos, tamvém conhecidos como Atributos e Códigos na forma de Procedimentos também conhecidos como Métodos. Vimos também que Objeto é uma abstração de algum fato ou ente do mundo real com atributos que representem as suas características ou propriedades e métodos que emulam o seu comportamento ou actividade, todas as propriedades e métodos comuns aos objetos encapsulam-se ou agrupam-se em classes onde a classe é um modelo para criar objeto, então diz-se que cada objeto é uma instância ou exemplar de uma classe.

CONCEITOS

**Abstração:** Consiste em um dos pontos mais importantes dentro de qualquer linguagem POO. Como trata-se de um objeto real, temos que imaginar o que esse objeto irá realizar dentro de nosso sistema. Nessa abstração devemos colocar em conta 3 pontos. 1° Dar **Identidade** ao objeto que iremos criar ( tem que única para não haver conflitos), 2° **Propriedades** (características do objeto) qualquer objeto possuí elementos que o definem dentro do POO. 3° **Métodos** degine as ações ou eventos que o objeto irá executar.

**Encapsulamento:** Trata-se de um dos elementos que adicionam segurança à aplicação em uma programação orientada a objetos pelo facto de esconder as propriedades, criando uma espécie de caixa preta.

**Herança:** É reuso de código, essa característica otimiza a produção da aplicação em tempo e linhas de códigos. O objeto da hierarquia irá herdar características de todos os códigos acima dele ( seus ancestrais), isso significa que o objeto pode herdar características de vários ancestrais ao mesmo tempo diretamente.

**Polimorfismo:** Consiste na alteração do funcionamento interno de um método herdado de um objeto pai. Denota uma situação na qual um objeto pode se comportar de maneiras diferentes ao receber uma mensagem.

**Interface:** É um recurso utilizado em JAVA para obrigar um determinado grupo de classes a ter métodos ou propriedades em comum para existir em determinado contexto. Os métodos podem ser implementados em cada classe de uma maneira diferente.

**R**EFERÊNCIAS **B**IBLIOGRÁFICAS

1 [www.davmedia.com.br](http://www.davmedia.com.br)

2 Wikipédia.com

.